POLYTECH DIJON

TRAVAUX PRATIQUES

SOUTIEN LINUX

Fusion des genres

OBJECTIFS

- Configurer Visual Studio Code pour une utilisation avec SSH
- Installer les éléments nécessaires à la compilation d'un programme C++
- Traduire un algorithme en C++

CONSIGNES

Vous rédigerez un compte-rendu personnel reprenant chacune des commandes nécessaires à la réalisation de ce sujet.

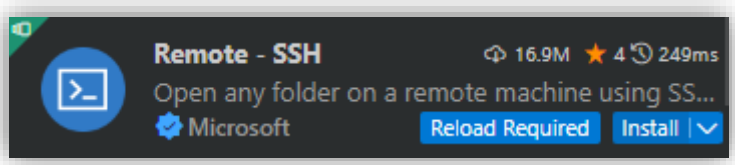
> **Astuce**
> Un compte-rendu numérique est recommandé afin de pouvoir facilement y copier vos commandes.

PREPARATION

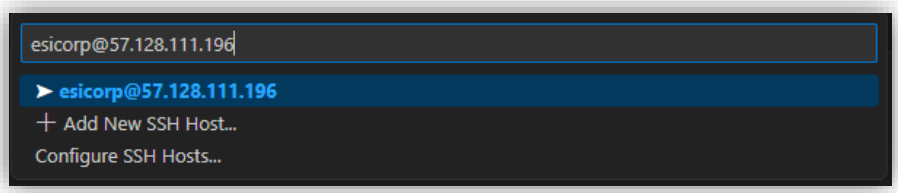
Visual Studio Code

Visual studio code est un éditeur de texte ++ très présent dans le monde actuel du développement. Il présente l'intérêt de proposer un grand nombre d'extensions qui augmentent considérablement ses fonctionnalités.

- Démarrez Visual Studio Code (pas Visual Studio !).
- Dans les extensions, sur la gauche de l'écran, installez « Remote SSH »

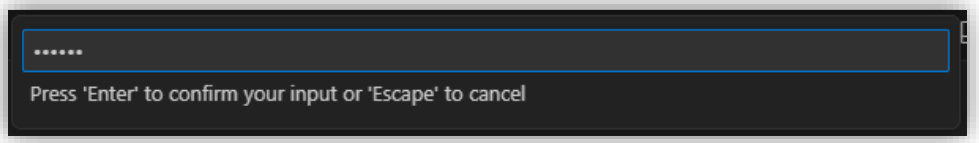


- Une fois installée, appuyez sur la touche « F1 » de votre clavier et sélectionnez « Remote SSH : Connect to Host… »
- Saisissez « esicorp@ip_machine » et validez avec la touche « Entrée » (l'ip de votre machine vous sera communiquée par votre enseignant).

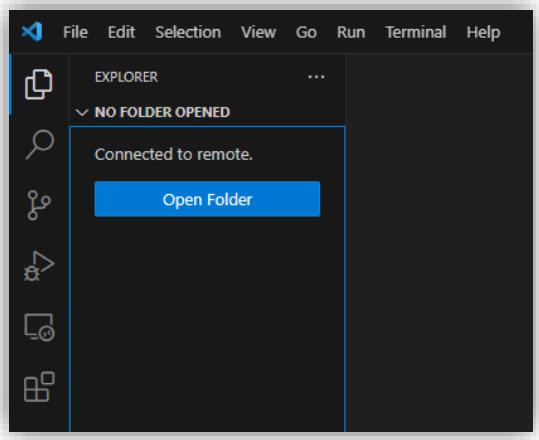


Une nouvelle fenêtre VS Code apparaît.

- Saisissez le mot de passe du compte SSH utilisé et validez avec « Entrée »

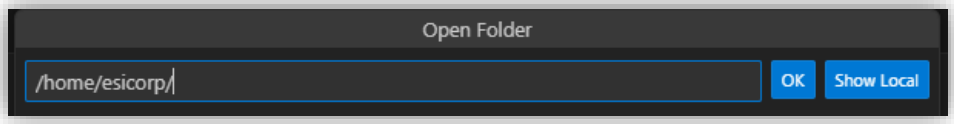


- Attendez que la connexion soit établie (statut en bas de page)
- Cliquez sur l'icône « Explorer » à gauche de l'écran :

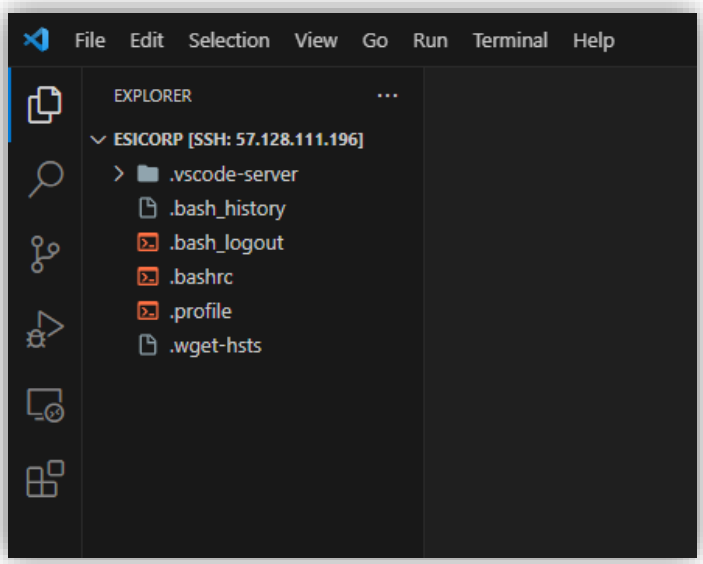


Vous devriez voir inscrit « Connected to remote »

- Cliquez sur « Open folder »
- Laissez le chemin par défaut « /home/esicorp » et cliquez sur le bouton « OK » :



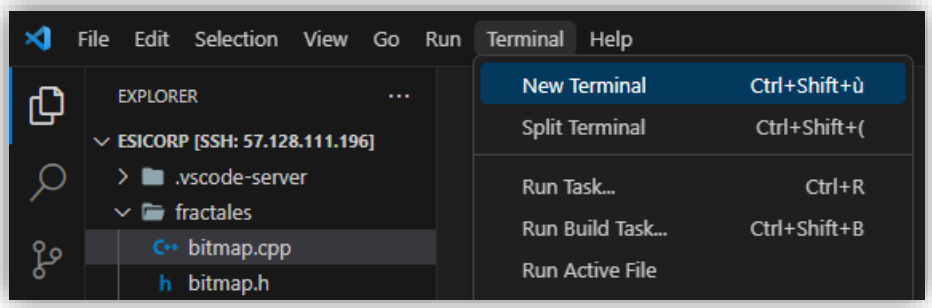
- Saisissez à nouveau le mot de passe utilisateur dans l'encadre en haut de la fenêtre.
- Attendez que VS code installe et initialise la connexion avec VS code server sur la machine distance (statut en bas de la fenêtre)
- Vous retrouvez, à gauche de l'écran, le contenu du dossier « /home/esicorp » distant :



- Faites un clic droit dans l'explorateur de fichiers et créez un nouveau dossier « fractales »

GCC

- Toujours sous Visual Studio Code, ouvrez un nouveau terminal :



Un terminal apparaît dans la partie inférieure de la fenêtre.

- Installez le paquet « **`build-essential`** » (n'oubliez pas de faire un **`update`** avant)

Petit test

- Créez un fichier « main.cpp » dans le dossier « fractales »
- Ecrivez le code permettant d'afficher « Cela semble fonctionner ! »
- Dans le terminal, compilez votre programme avec la commande suivante

```
gcc main.cpp -o program -lstdc++
```

- Dans le terminal, exécutez votre programme avec la commande suivante

```
./program
```

Si vous voyez apparaître « Cela semble fonctionner !», c'est que cela fonctionne 😊

BITMAP

Récupération des fichiers

- Récupérez les fichiers « bitmap.h » et « bitmap.cpp » sur https://lamarmotte.info
- Glissez/déposez les fichiers téléchargés dans le dossier « fractales » dans l'interface de VS Code. Les fichiers seront directement téléversés sur la machine distante.

La classe **Bitmap**, déclarée dans le fichier bitmap.h, permet de créer une image et de définir la couleur de chacun de ses pixels avant de l'enregistrer au format bitmap.

Un point

- Modifiez le code du fichier main.cpp précédent à partir du code suivant :

```
#include "bitmap.h"

int main()
{
    //Crée une image de 200 pixels de large et de 100 pixels de haut
    Bitmap image(200, 100);

    //Place un point rouge au centre de l'image (x=100, y=50)
    image.setPixelColor(100, 50, 255, 0, 0);

    //Enregistre l'image dans le fichier "test.bmp"
    image.save("test.bmp");

    return 0 ;
}
```

- Dans le code ci-dessus, idenfiez ce qui définit la couleur du pixel (ici rouge).

- Compilez le programme à l'aide de la commande suivante :

```
gcc main.cpp bitmap.cpp -o program -lstdc++
```

- Exécutez le programme

- Regardez dans l'explorateur de fichiers de VS Code, un fichier « test.bmp » a dû apparaître.
- Ouvrez-le dans VS Code.

Un carré

- En vous inspirant du code précédent, dessinez un carré violet de 50 pixels de côté.
- Testez le bon fonctionnement.

MANDELBROT

- Dans la fonction main du fichier précédent, traduisez l'algorithme suivant en C++ :

```
définir x1 = -2.1
définir x2 = 0.6
définir y1 = -1.2
définir y2 = 1.2
définir zoom = 100
définir iteration_max = 50

définir image_width = (x2 - x1) * zoom
définir image_height = (y2 - y1) * zoom

Pour x = 0 tant que x < image_width par pas de 1
     Pour y = 0 tant que y < image_height par pas de 1
         définir c_r = x / zoom + x1
         définir c_i = y / zoom + y1
         définir z_r = 0
         définir z_i = 0
         définir i = 0

         Faire
             définir tmp = z_r
             z_r = z_r*z_r - z_i*z_i + c_r
             z_i = 2*z_i*tmp + c_i
             i = i+1
         Tant que z_r*z_r + z_i*z_i < 4 et i < iteration_max

         Si i = iteration_max
             dessiner en noir le pixel de coordonnée (x ; y)
         Sinon
             dessiner avec couleur rgb(0, 0, i * 255 / iteration_max)
             le pixel de coordonnée (x ; y)
         finSi
     finPour
 finPour
```

Algorithme extrait de https://zestedesavoir.com/tutoriels/pdf/329/dessiner-la-fractale-de-mandelbrot.pdf

- N'oubliez pas d'enregistrer l'image.
- Contemplez votre œuvre.

ENCORE UN PEU DE TEMPS ?

VIM

- Réalisez le tutoriel de l'éditeur VIM en exécutant la commande suivante :

```
vimtutor
```

Plus de fractales

- Traduisez d'autres algorithmes de fractales à partir du document suivant : *https://zestedesavoir.com/tutoriels/pdf/329/dessiner-la-fractale-de-mandelbrot.pdf*